

RESEARCH

Open Access



CMIP: a software package capable of reconstructing genome-wide regulatory networks using gene expression data

Guangyong Zheng^{1*†}, Yaochen Xu^{1,2†}, Xiujun Zhang³, Zhi-Ping Liu³, Zhuo Wang⁴, Luonan Chen^{3*} and Xin-Guang Zhu^{1*}

From The 27th International Conference on Genome Informatics
Shanghai, China. 3-5 October 2016

Abstract

Background: A gene regulatory network (GRN) represents interactions of genes inside a cell or tissue, in which vertexes and edges stand for genes and their regulatory interactions respectively. Reconstruction of gene regulatory networks, in particular, genome-scale networks, is essential for comparative exploration of different species and mechanistic investigation of biological processes. Currently, most of network inference methods are computationally intensive, which are usually effective for small-scale tasks (e.g., networks with a few hundred genes), but are difficult to construct GRNs at genome-scale.

Results: Here, we present a software package for gene regulatory network reconstruction at a genomic level, in which gene interaction is measured by the conditional mutual information measurement using a parallel computing framework (so the package is named CMIP). The package is a greatly improved implementation of our previous PCA-CMI algorithm. In CMIP, we provide not only an automatic threshold determination method but also an effective parallel computing framework for network inference. Performance tests on benchmark datasets show that the accuracy of CMIP is comparable to most current network inference methods. Moreover, running tests on synthetic datasets demonstrate that CMIP can handle large datasets especially genome-wide datasets within an acceptable time period. In addition, successful application on a real genomic dataset confirms its practical applicability of the package.

Conclusions: This new software package provides a powerful tool for genomic network reconstruction to biological community. The software can be accessed at <http://www.picb.ac.cn/CMIP/>.

Keywords: Gene regulatory network, Genome-wide, Parallel computing, Software

* Correspondence: zhenggy@sibs.ac.cn; lnchen@sibs.ac.cn;
zhuxingguang@picb.ac.cn

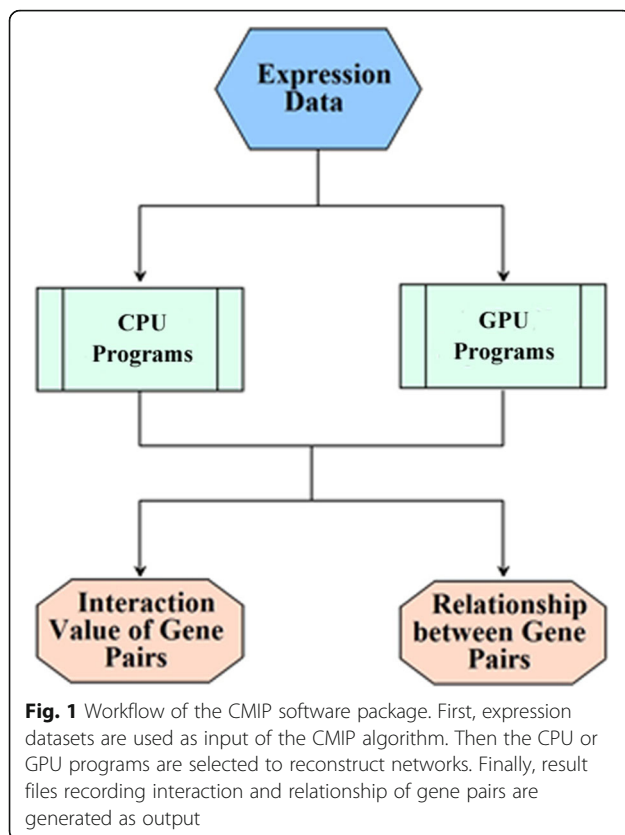
†Equal contributors

¹CAS Key Laboratory of Computational Biology and State Key Laboratory of Hybrid Rice, CAS-MPG Partner Institute for Computational Biology, Chinese Academy of Sciences, 320 Yueyang Road, Shanghai 20031, China

³Key Laboratory of Systems Biology, Institute of Biochemistry and Cell Biology, Chinese Academy of Sciences, 320 Yueyang Road, Shanghai 200031, China

Full list of author information is available at the end of the article





Background

In the post-genome era, an important task of molecular biology is to reconstruct gene regulatory networks (GRNs), which represent interactions between genes inside a cell or tissue. A GRN provides molecular interactions and regulatory effects of components involved in a biological process, and hence provides insights into the molecular mechanism of the process [1, 2]. In detail, GRNs can be used to interpret biological processes through studying topological structure information of sub-networks related to these processes, where genes facilitate specific biological functions together [3, 4]. GRNs can help annotate genes clustered in modules and motifs since genes in the same module or motif have similar functions [5, 6]. GRNs can be utilized to identify dynamical network biomarkers (DNB) at the critical states of biological processes if stage-wise data are available, which help biologists understand mechanism of biological process better [7, 8]. Therefore, reconstruction of GRNs can not only support investigating roles of genes and components involved in a biological process, but also help study how a process is developed and maintained.

In the last decade, many algorithms have been developed to infer GRNs based on reverse-engineering methods, such as Bayesian network [9–11], Boolean network [12, 13], linear and non-linear regression [14–18],

differential equation [19, 20], information-theoretic approaches [21–26], probabilistic phylogeny network [27], part mutual information network [28], and probabilistic graphical models [29–32]. In 2011, we proposed a GRN inference algorithm, named PCA-CMI, which can distinguish direct interactions of gene pairs from indirect ones based on the conditional mutual information (CMI) measurement [33–35]. However, two limitations of the algorithm hinder its wide application. One is that an appropriate threshold should be assigned to the method for direct interactions judgment in advance, which is difficult for users since the threshold is hard to select before GRN reconstruction. The other is that the method is time-costly especially for genomic network reconstruction, which is a common restriction of most current GRN inferring methods.

In this report, we describe a new software package CMIP, which implements the PCA-CMI algorithm with the goal of enable biologists to build genomic networks easily. The CMIP package incorporates a threshold determination method and a parallel computing process for network inference. The threshold determination method can choose an appropriate cutoff on-the-fly for gene interaction judgment. Computing procedure of the CMI measurement is optimized to make the algorithm robust, in which parallel computing strategies are applied to accelerate calculation process. This paper describes the algorithm details, program implementation, prediction performance, and practical application of the CMIP package.

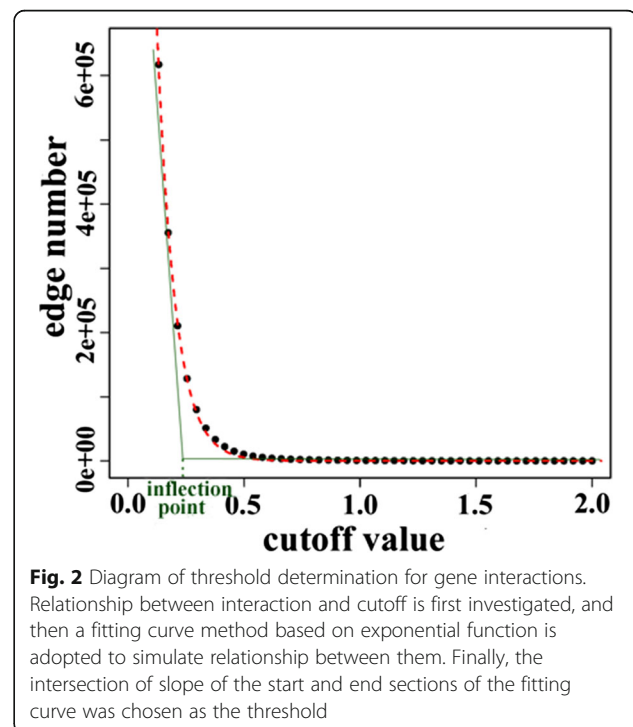


Table 1 Effectiveness of threshold determination method under different criteria

| Criteria/manners | 0-order | 1-order | 2-order | 3-order | Percentage |
|----------------------|---------|---------|---------|---------|------------|
| Offset less than 5% | 3 | 7 | 4 | 4 | 45% |
| Offset less than 10% | 5 | 9 | 9 | 7 | 75% |
| Offset less than 20% | 7 | 10 | 10 | 10 | 93% |

Methods

Workflow of the CMIP package

As showed in Fig. 1, the CMIP package first uses an expression data file as input, in which expression value of genes under different experimental conditions are provided. Then, CMIP program calculates the correlation value between gene pairs. During calculation process, a threshold determination method is called to generate an appropriate cutoff for direct interaction judgment. When the process is finished, two result files are produced as output. One is a gene interaction file, recording raw correlation value of gene pairs. The other is a gene relation file, providing the relationship between gene pairs. In practice, relationship of a gene pair is assigned as 1 when their correlation value is over the interaction cutoff. Otherwise corresponding value is assigned as 0.

Correlation calculation of the CMIP algorithm

The algorithm implemented in the CMIP package is as follows. First, correlation values of each gene pairs are calculated using the mutual information (MI) measurement. Then a threshold determination method (described in the “Threshold determination of gene interaction” section) is called to provide an appropriate interaction cutoff for gene pairs. An interaction is

marked for gene X and Y when their raw correlation value is over the cutoff. After that, for each gene interaction, their correlation values are updated through calculating the conditional mutual information (CMI) measurement (Eq. 1–4), which describes the dependence of two genes given neighboring genes as condition. A gene Z is defined as a neighbor of gene X and Y when it has interactions with both gene X and Y. In practice, the maximum CMI value between gene X and Y is kept. Finally, for gene X and Y, they are regarded as having direct interaction when their CMI value is over the interaction cutoff and their relationship value is set to be 1 as output.

$$I(X, Y|Z) = \sum_{x \in X, y \in Y, z \in Z} p(x, y, z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} \quad (1)$$

Where $I(X, Y|Z)$ is CMI measurement between gene X and Y given gene Z as a condition; $p(x, y, z)$ are joint probability of gene triple (X, Y, Z); while $p(x|z)$, $p(y|z)$, and $p(x, y|z)$ are conditional probabilities of gene X, Y, and gene pair (X, Y) given gene Z as a condition. According to information theory, the CMI measurement can also be defined as follows.

$$H(X, Y|Z) = H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z) \quad (2)$$

Where $H(Z)$ is the entropy of gene Z; $H(X, Z)$, $H(Y, Z)$ and $H(X, Y, Z)$ are joint entropies of gene pair (X, Z), (Y, Z) and gene triple (X, Y, Z); $H(X, Y|Z)$ is the conditional entropy of genes X and Y given gene Z as a condition. Based on the Gaussian distribution, the entropy of gene Z can be estimated as follows.

$$H(Z) = \log \left[(2\pi e)^{n/2} |C(Z)|^{1/2} \right] = \frac{1}{2} \log (2\pi e)^n |C(Z)| \quad (3)$$

Where n is number of experiment, $C(Z)$ is the covariance matrix of gene Z, and $|C(Z)|$ is the determinant of the matrix. While joint entropies can be estimated similarly through corresponding covariance matrixes. Based on the entropy estimator, in practice the CMI measurement (I) is calculated as follows.

$$I(X, Y|Z) = \frac{1}{2} \log \frac{|C(X, Z)| \cdot |C(Y, Z)|}{|C(Z)| \cdot |C(X, Y, Z)|} \quad (4)$$

Where $C(X, Z)$ and $C(Y, Z)$ are covariance matrixes of gene pair (X, Z) and (Y, Z); $C(X, Y, Z)$ is covariance matrix of gene triple (X, Y, Z); the $|C|$ is determinant of a matrix.

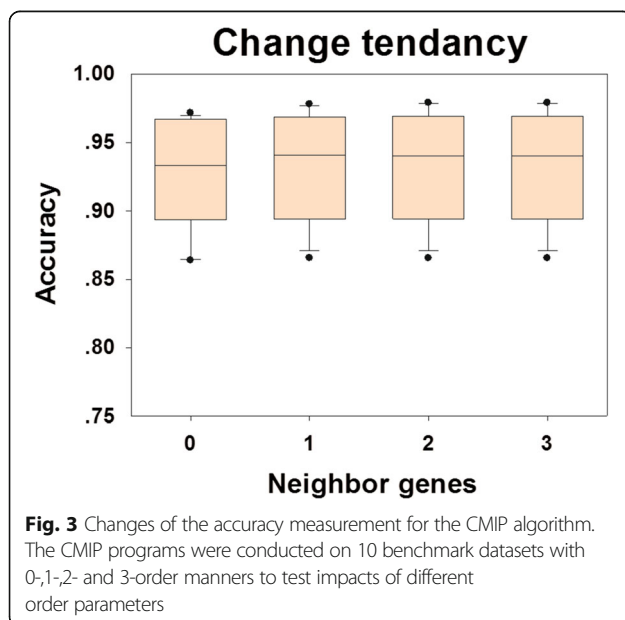


Table 2 Scores of various network inference methods on benchmark datasets

| Measurements | ARACNE | CLR | MI | GENIE3 | Inferelator | TIGRESS | CMIP |
|--------------|--------|--------|--------|--------|-------------|---------|--------|
| AUROC | 0.6689 | 0.7632 | 0.8004 | 0.7955 | 0.7014 | 0.8048 | 0.7945 |
| AUPR | 0.1465 | 0.2076 | 0.3537 | 0.3033 | 0.2172 | 0.3991 | 0.3637 |
| Average | 0.4077 | 0.4854 | 0.5771 | 0.5494 | 0.4593 | 0.6020 | 0.5791 |

Threshold determination of gene interaction

Given interaction of gene pairs, the number of interactions decreases dramatically with the increase of the cut-off and their relationship shows an exponential decay. Therefore, in practice we chose to use an exponential function to simulate relationship between interaction and cutoff. Correlation values of gene pairs are first calculated as mentioned in the “Correlation calculation of the CMIP algorithm” section. Then direct interactions between gene pairs under different cutoffs are estimated and a scatter plot is generated (Fig. 2), where X axis is the cutoff value and Y axis is the number of direct interactions. After that, we fit the number of direct interactions as a function of the cutoff value with an exponential function. Finally, we chose the threshold as the intersection of slope of the start and end sections of the fitting curve, which represents the inflection point of the curve.

Parallelization of the CMIP programs

In CMIP, parallel strategies were applied to speed up computing process of correlation. In practice, a CPU and a GPU version program of CMIP algorithm were

developed so that users could utilize them in different computational environment. The CPU version program is implemented based on the OpenMP framework [36], where loop calculation is accelerated with the multi-threads technology. In detail, the total computing task of correlation is first calculated based on gene numbers, and then computing tasks is partitioned equally to each CPU node. While the GPU version program is implemented based on the CUDA framework [37], where serial and parallel computing tasks are undertaken by CPU and GPU cores respectively. In detail, a production-consumption strategy is used in the GPU version program, in which gene expression data used by correlation calculation is first processed by the CPU cores (production); then pre-processed data is delivered to GPU cores for correlation calculation (consumption) using a parallel mode; finally, the results are transferred from GPU to CPU cores for aggregation.

Evaluation of network inference methods

Receiver operating characteristic (ROC) curve and precision-recall (PR) curve are used to evaluate performance of different network inference methods. The ROC

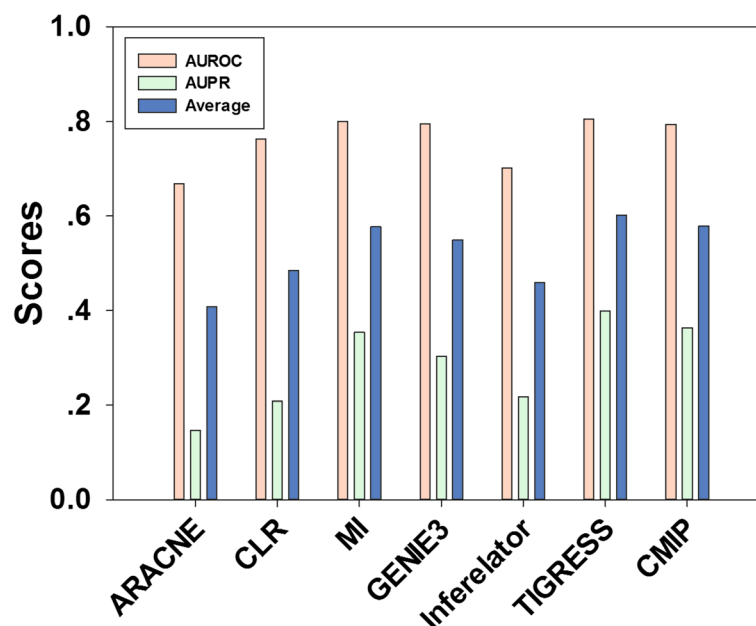


Fig. 4 Scores of various network inference methods. Performance of various network inference methods are compared on 10 benchmark datasets using the AUROC and AUPR measurement

Table 3 Running time of different network inference programs

| Methods | CMIP ^a | CMIP ^b | MI | ARACNE | CLR | GENIE3 | Inferelator | TIGRESS |
|----------------|-------------------|-------------------|----|--------|-----|--------|-------------|---------|
| Multi-threads | Yes | Yes | No | No | No | No | Yes | No |
| Time (seconds) | 7 | 4 | 8 | 353 | 68 | 4715 | 2728 | 13089 |

^aGPU version program, ^b CPU version program

curve is a graphical plot which illustrates discrimination capacity of algorithm under various thresholds for binary classifier problems, where the X and Y axis are false and true positive rate respectively. While the PR curve shows recognition capability of algorithm under various thresholds for positive samples, in which the X and Y axis represent the recall and precision measurement respectively. Commonly, area under the ROC curve (AUROC) and area under the PR curve (AUPR) are calculated to comprehensively evaluate performance of a network inference method. In practice, the true positive rate (TPR, also known as recall), false positive rate (FPR), and positive predictive value (PPV, also known as precision) and accuracy (ACC) are calculated as follows.

$$\begin{aligned} TPR &= TP / (TP + FN) \\ FPR &= FP / (TN + FP) \\ PPV &= TP / (TP + FP) \\ ACC &= (TP + TN) / (TP + FN + TN + FP) \end{aligned}$$

(5)

Where TP, TN, FP and FN are numbers of true positive, true negative, false positive and false negative respectively. Given a true interaction between genes X and Y, it is recorded as a true positive item if it is predicted by the algorithm. Otherwise, it is recorded as a false negative item. Similarly, for a non-interaction gene pair X and Y, it is recorded as a false positive item when predicted by the algorithm. Otherwise, it is recorded as a true negative item.

Results and discussion

Efficiency of the threshold determination method

In this study, we developed an automatic threshold determination method for interaction cutoff prediction. Here we show the efficiency of the method using numerical experiments. First, 10 benchmark datasets (synthetic datasets) were collected from the DREAM3 competition website [38]. Then, we ran CMIP programs on these datasets with 0-, 1-, 2-, or 3-order manner separately, which means the CMI is calculated given 0, 1, 2, or 3 neighboring genes as conditions. Note that when no

Table 4 Running time of the CMIP programs in pineapple GRNs reconstruction

| Programs/time (seconds) | Leaf base network | Leaf tip network |
|-------------------------|-------------------|------------------|
| CMIP ^a | 1828 | 2277 |
| CMIP ^b | 2026 | 2733 |

^aGPU version program, ^b CPU version program

neighboring gene is given as condition, the CMI measurement is equivalent to the MI measurement. In practice, the CMIP programs were run with a predefined cutoff, which was increased from 0 to 1 with a step size of 0.02. For each running of programs, accuracy of the CMIP algorithm under a certain cutoff was recorded. After that, accuracies under different cutoffs were checked, and the cutoff at which corresponding accuracy measure reached its maximum was stored as the true threshold. On the other hand, the CMIP programs were run with the automatic threshold determination method (see the “Threshold determination of gene interaction” section for details) and a predicted threshold was presented. Finally, offset of threshold was detected through comparing the true and predicted threshold values (Eq. 6).

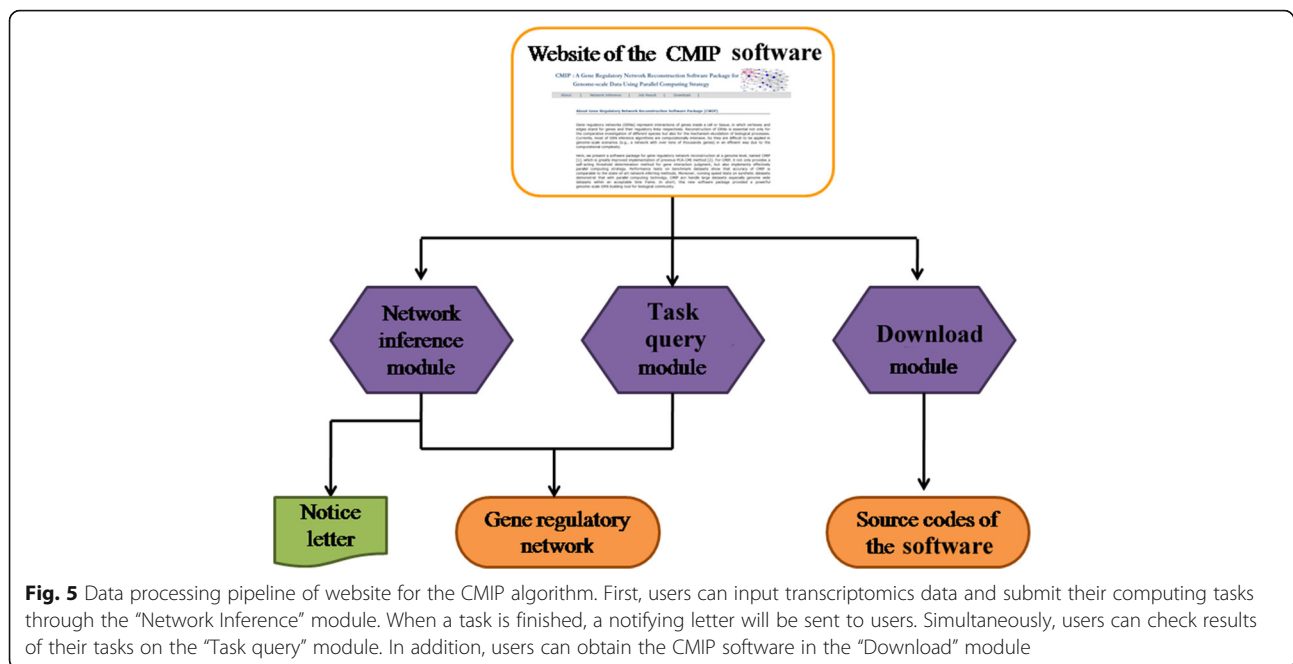
$$offset = \frac{|true\ threshold - predicted\ threshold|}{maximum - minimum} \times 100\%$$

(6)

In our work, the stringent, standard, and moderate criteria are defined as offset less than 5, 10, and 20% respectively. Results of offset detection are shown in Table 1, where each cell represents the number of datasets for which the offsets satisfy a certain criteria under a defined order (0-, 1-, 2-, or 3-order). Totally, the proportion of datasets that satisfies the stringent, standard, and moderate criteria are 45, 75, and 93% respectively. These results demonstrate that the new threshold determination method developed in this study is effective and an appropriate cutoff can be provided on-the-fly during calculation process of correlation. Though in this study, we already include 10 datasets to test the threshold determination method, it is possible that there are networks for which the current threshold determination method might be not the best option; therefore, development of new methods for threshold determination is still needed in the future.

Parameter selection of the CMIP software

The CMIP algorithm is based on the CMI measurement, where computational complexity increases exponentially with the increase of the number of neighboring genes as conditions, i.e. with the increase of orders. So an appropriate order parameter needs to be selected for the algorithm. Here, we tested the impacts of different order parameters on prediction accuracy. In practice, 10 synthetic benchmark datasets were first collected from the DREAM3 website [38]. Then the CMIP programs were



run on these datasets with 0-, 1-, 2-, and 3-order manners separately and the accuracy measurement was inspected in each running. Inspection results show that the accuracy of the CMIP algorithm gradually increased with the increase of the order parameter from 0 to 3; however, the increasing trend becomes flat after the order of 1 (Fig. 3). Considering both the accuracy and computational complexity, we recommend setting 1 to be the order parameter for the CMIP algorithm.

Performance evaluation of the CMIP package on benchmark datasets

The CMIP package was utilized to reconstruct GRNs on 10 DREAM3 benchmark datasets for performance evaluation. In practice, programs of the package were run with a 1-order manner, i.e. the CMI measurement of gene pair was calculated given one neighbor gene as condition. Subsequently, running results of CMIP were compared with other popular network inference methods using the AUROC and AUPR measurements. Specifically, programs of popular GRN inference methods were downloaded from website of the DREAM projects [39] and recommended parameters were used during running these programs. Mean scores of the AUROC and AUPR measurement on 10 benchmark datasets for various methods are shown in Table 2 and Fig. 4. Given both the AUROC and AUPR measurements, the CMIP algorithm achieves high performance and delivers competitive values to popular methods. For average score of the AUROC and AUPR measurement, the CMIP algorithm performs better than all methods except the TIGRESS algorithm. These results demonstrate that the

CMIP algorithm is comparable to most currently used network inference methods.

Application of the CMIP package on real biological datasets

We further applied the CMIP software on real transcriptome data to check its practical applicability. The CMIP software was first used to build GRNs of pineapple leaves. In detail, a GRN of leaf base and a GRN of leaf tip were constructed based on genome-scale expression data. Totally, 15,483 genes (201,537 interactions) and 13,543 genes (188,391 interactions) were included in the base and tip GRNs respectively. Analysis of the node degree distribution suggested that both the tip and the base network showed small-world properties. Then, we extracted genes linked to metabolic enzymes of Crassulacean Acid Metabolism (CAM) in the base and tip networks. After that, genes linked to metabolic enzymes in the tip network but missed in the base network were identified as potential recruited regulators of CAM photosynthesis. Subsequently experimental study showed that regulators identified from network comparison do play important roles in photosynthesis differentiation [40]. This application of CMIP software on real dataset shows its effectiveness and efficiency for genomic GRNs reconstruction.

Effectiveness of parallel computing framework of CMIP programs

Since the CMIP software is developed to infer GRNs for genome-scale datasets, parallel computation strategies are adopted in the software to speed up computing

process. Here, we compared running time of the CMIP programs with popular network inference methods. In practice, CPU version programs were run on a Linux computing server, which has two Intel Xeon E5-2650 CPU units containing 32 CPU cores in total. While GPU version programs were run on a Linux computing server, which has 2 NVIDIA Tesla k20m GPU cards containing 4992 CUDA cores in total. In this study, all program was executed on a synthetic dataset (collected from published literatures), which included expression values of 500 genes under 343 biological treatments for Arabidopsis. Running times of different programs are shown in Table 3. Programs of the CMIP algorithm are much faster than most popular methods. Running time of the CMIP software applied in pineapple GRNs reconstruction is shown in Table 4. These results suggest that parallel computing strategies applied in the CMIP software are efficient and the software can handle genome-scale datasets within a reasonable time period.

Usage of the CMIP package

A service website of the CMIP programs is established so that users can utilize them remotely (Fig. 5). Now, the website can be accessed at <http://www.picb.ac.cn/CMIP/>. The web service is created based on a remote resource management system. Once a task is submitted to the system, calculation resources include CPU and GPU computing components will be automatically assigned. To use the CMIP programs, users first need to submit their computing tasks through inputting expression data on the “Network inference” module. Then tasks are handled by the management system. When a task is finished, a notice letter will be sent to users. Alternatively, users can query status of their tasks through the “Job Result” module. Finally, summary information of the task will be presented on the website. In addition to the online manners, users can download the source codes of CMIP software from the “Download” module (Additional file 1), and then use it on local computing servers.

Conclusions

In this study, we provide a new software package for network inference, which can reconstruct genomic GRNs within a short time period. The software package has a number of novel features compared with other GRN inference methods. First, CMIP can detect direct gene interactions from indirect ones with a high accuracy based on the CMI measurement. Results of performance evaluation on benchmark datasets show that precision and accuracy of the CMIP algorithms are comparable to most currently used methods. Secondly, an automatic threshold determination method is incorporated into the CMIP algorithm, so users do not need specify a

predefined cutoff for gene interaction judgment and an appropriate threshold can be provided on-the-fly. Numerical experiments confirm the efficiency of the threshold determination method. Last but not least, the OpenMP and CUDA framework are applied in the software to speed up computing process of the CMIP algorithms, which enables the software to build GRNs with less running time. With this feature, the software is suitable to reconstruct genomic GRNs. The area of CMIP that needs future development is that it can't provide directionality to edges of gene regulatory networks, which is a common limitation of many current methods, such as CLR [24] and minet [22]. This limitation can be resolved by a two-steps routine. First, using the CMIP software to build a gene regulatory network as background model, then giving directionality to edges of the network according to results of biochemical perturbation experiments, or predicting directionality for edges of the network based on time series expression data [41].

Additional file

Additional file 1: S1- CMIP_code.zip: source codes of the CMIP software package. (ZIP 46 kb)

Acknowledgments

We thank the anonymous reviewers for their constructive suggestion that help improve quality of the manuscript.

Declarations

This article has been published as part of BMC Bioinformatics Volume 17 Supplement 17, 2016: Proceedings of the 27th International Conference on Genome Informatics: bioinformatics. The full contents of the supplement are available online at <http://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-17-supplement-17>.

Funding

This study was supported by the Shanghai Municipal Natural Science Foundation [grant number 14ZR1446700], SA-SIBS Scholarship Program, National 863 Program Green Super Rice [grant number 2014AA101601], National Basic Research and Development Plan of China [grant number: 2015CB150104], CAS Strategic Research Project [grant number XDA08020301], and the National Natural Science Foundation of China [grant numbers 91529303, 91439103, 61134013, 81471047]. Publication costs for this study were funded by foundation mentioned above.

Availability of data and material

The datasets analyzed during the current study are available in the DREAM3 repository <http://dreamchallenges.org/project-list/dream3-2008/>.

Authors' contributions

GYZ, LNC, XJZ, ZPL, ZW, and XGZ conceived and designed the experiments. GYZ and YCX performed the experiments and analyzed the data. GYZ, LNC and XGZ wrote the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Author details

¹CAS Key Laboratory of Computational Biology and State Key Laboratory of Hybrid Rice, CAS-MPG Partner Institute for Computational Biology, Chinese Academy of Sciences, 320 Yueyang Road, Shanghai 20031, China. ²Software Engineering Institute, East China Normal University, 3663 North Zhongshan Road, Shanghai 200062, China. ³Key Laboratory of Systems Biology, Institute of Biochemistry and Cell Biology, Chinese Academy of Sciences, 320 Yueyang Road, Shanghai 200031, China. ⁴College of Life Science and Biotechnology, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, China.

Published: 23 December 2016

References

- Barabasi AL, Oltvai ZN. Network biology: understanding the cell's functional organization. *Nat Rev Genet.* 2004;5(2):101–13.
- Gardner TS, di Bernardo D, Lorenz D, Collins JJ. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science.* 2003;301(5629):102–5.
- Artzy-Randrup Y, Fleishman SJ, Ben-Tal N, Stone L. Comment on "Network motifs: simple building blocks of complex networks" and "Superfamilies of evolved and designed networks". *Science.* 2004;305(5687):1107. author reply 1107.
- Braha D, Bar-Yam Y. Topology of large-scale engineering problem-solving networks. *Phys Rev E Stat Nonlin Soft Matter Phys.* 2004;69(1 Pt 2):016113.
- Angeli D, Ferrell Jr JE, Sontag ED. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc Natl Acad Sci U S A.* 2004;101(7):1822–7.
- Ma W, Trusina A, El-Samad H, Lim WA, Tang C. Defining network topologies that can achieve biochemical adaptation. *Cell.* 2009;138(4):760–73.
- Chen L, Liu R, Liu ZP, Li M, Aihara K. Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers. *Sci Rep.* 2012;2:342.
- Liu R, Wang X, Aihara K, Chen L. Early diagnosis of complex diseases by molecular biomarkers, network biomarkers, and dynamical network biomarkers. *Med Res Rev.* 2013;34(3):455–78.
- Liu F, Zhang SW, Guo WF, Wei ZG, Chen L. Inference of Gene Regulatory Network Based on Local Bayesian Networks. *PLoS Comput Biol.* 2016;12(8):e1005024.
- Zou M, Conzen SD. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics.* 2005;21(1):71–9.
- Brown LE, Tsamardinos I, Aliferis CF. A novel algorithm for scalable and accurate Bayesian network learning. *Stud Health Technol Inform.* 2004; 107(Pt 1):711–5.
- Kauffman S, Peterson C, Samuelsson B, Troein C. Random Boolean network models and the yeast transcriptional network. *Proc Natl Acad Sci U S A.* 2003;100(25):14796–9.
- Grieb M, Burkovski A, Strang JE, Kraus JM, Gross A, Palm G, Kuhl M, Kestler HA. Predicting Variabilities in Cardiac Gene Expression with a Boolean Network Incorporating Uncertainty. *PLoS One.* 2015;10(7):e0131832.
- Haury AC, Mordelet F, Vera-Licona P, Vert JP. TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. *BMC Syst Biol.* 2012;6:145.
- Wang Y, Joshi T, Zhang XS, Xu D, Chen LN. Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics.* 2006;22(19): 2413–20.
- Marbach D, Costello JC, Kuffner R, Vega NM, Prill RJ, Camacho DM, Allison KR, Kellis M, Collins JJ, Stolovitzky G. Wisdom of crowds for robust gene network inference. *Nat Methods.* 2012;9(8):796–804.
- Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, Thorsson V. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.* 2006;7(5):R36.
- Brooks AN, Reiss DJ, Allard A, Wu WJ, Salvanha DM, Plaisier CL, Chandrasekaran S, Pan M, Kaur A, Baliga NS. A system-level model for the microbial regulatory genome. *Mol Syst Biol.* 2014;10:740.
- Cantone I, Marucci L, Iorio F, Ricci MA, Belcastro V, Bansal M, Santini S, di Bernardo M, di Bernardo D, Cosma MP. A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell.* 2009; 137(1):172–81.
- Honkela A, Girardot C, Gustafson EH, Liu YH, Furlong EE, Lawrence ND, Rattray M. Model-based method for transcription factor target identification with limited data. *Proc Natl Acad Sci U S A.* 2010;107(17):7793–8.
- Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics.* 2006;7 Suppl 1:S7.
- Meyer PE, Lafitte F, Bontempi G. minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics.* 2008;9:461.
- Yu X, Zheng G, Shan L, Meng G, Vingron M, Liu Q, Zhu XG. Reconstruction of gene regulatory network related to photosynthesis in *Arabidopsis thaliana*. *Front Plant Sci.* 2014;5(3):273.
- Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS. Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.* 2007;5(1):e8.
- Usadel B, Obayashi T, Mutwil M, Giorgi FM, Bassel GW, Tanimoto M, Chow A, Steinhauser D, Persson S, Provart NJ. Co-expression tools for plant biology: opportunities for hypothesis generation and caveats. *Plant Cell Environ.* 2009;32(12):1633–51.
- Chevalier M, Venturelli O, El-Samad H. The Impact of Different Sources of Fluctuations on Mutual Information in Biochemical Networks. *PLoS Comput Biol.* 2015;11(10):e1004462.
- Zhang X, Moret BM. Refining regulatory networks through phylogenetic transfer of information. *IEEE/ACM Trans Comput Biol Bioinform.* 2012;9(4): 1032–45.
- Zhao J, Zhou Y, Zhang X, Chen L. Part mutual information for quantifying direct associations in networks. *Proc Natl Acad Sci U S A.* 2016;113(18):5130–5.
- Weaver DC, Workman CT, Stormo GD. Modeling regulatory networks with weight matrices. *Pac Symp Biocomput.* 1999;112–123.
- Kramer N, Schafer J, Boulesteix AL. Regularized estimation of large-scale gene association networks using graphical Gaussian models. *BMC Bioinformatics.* 2009;10:384.
- Friedman N. Inferring cellular networks using probabilistic graphical models. *Science.* 2004;303(5659):799–805.
- Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P. Inferring regulatory networks from expression data using tree-based methods. *PLoS One* 2010, 5(9).
- Zhang X, Liu K, Liu ZP, Duval B, Richer JM, Zhao XM, Hao JK, Chen L. NARROM: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference. *Bioinformatics.* 2012;29(1):106–13.
- Zhang X, Zhao J, Hao JK, Zhao XM, Chen L. Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks. *Nucleic Acids Res.* 2014;43:e31.
- Zhang X, Zhao XM, He K, Lu L, Cao Y, Liu J, Hao JK, Liu ZP, Chen L. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics.* 2011; 28(1):98–104.
- Rabenseifner R, Hager G, Jost G. Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes. In: Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on: 18–20 Feb. 2009. 2009. p. 427–36.
- Nickolls J, Buck I, Garland M, Skadron K. Scalable Parallel Programming with CUDA. *Queue.* 2008;6(2):40–53.
- Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci U S A.* 2010;107(14):6286–91.
- Marbach D, Schaffter T, Mattiussi C, and Floreano D. Generating Realistic "in silico" Gene Networks for Performance Assessment of Reverse Engineering Methods. *J Computational Biol.* 2009;16(2):229–239.
- Ming R, VanBuren R, Wai CM, Tang H, Schatz MC, Bowers JE, Lyons E, Wang ML, Chen J, Biggers E, et al. The pineapple genome and the evolution of CAM photosynthesis. *Nat Genet.* 2015;47(12):1435–42.
- Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, Califano A. Reverse engineering cellular networks. *Nat Protoc.* 2006;1(2):662–71.